THIS MANUAL IS CURRENT
FOR MDBS 3.09

MDBS III

RECOVERY AND TRANSACTION LOGGING

REFERENCE MANUAL

VERSION 3.09

MARCH 1987

Micro Data Base Systems, Inc.

P. O. Box 248

Lafayette, Indiana 47902

USA

Telex: 5106017487 (MDBS LAF UQ)

(800) 344-5832

Copyright Notice

This entire manual is provided for the use of the customer and the customer's employees. The entire contents have been copyrighted by Micro Data Base Systems, Inc., and reproduction by any means is prohibited except as permitted in a written agreement with Micro Data Base Systems, Inc.

© Copyright 1987 Micro Data Base Systems, Inc.

3251916

NEW VERSIONS, LICENSING, AND DISCLAIMER

Any programming endeavor of the magnitude of the MDBS III software continually evolves over time. Realizing this, Micro Data Base Systems, Inc. (**mdbs**), provides updates to this product as part of a service and maintenance program.

New versions of MDBS III software are considered separate products; however, bona fide owners of previous versions are generally entitled to a preferential rate structure.

Each copy of our software is personalized to identify the licensee. There are several levels of this personalization, some of which involve encryption methods guaranteed to be combinatorially difficult to decipher. Our products have been produced with a very substantial investment of capital and labor, to say nothing of the years of prior involvement in the data base management area by our principals. Accordingly, we are seriously concerned about any unauthorized copying of our products and will take any and all available legal action against illegal copying or distribution of our products.

All rights reserved. No part of this material shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from **mdbs**.

Although care has been taken in the preparation of this material, **mdbs** assumes neither responsibility for errors or omissions, nor for damages resulting from the use of the information contained herein. **mdbs** does not warrant its accuracy nor guarantee the operation of the system in every instance described herein.

The reader/user assumes full liability and all risk for any damages resulting from the use of the information contained herein, and for determining whether the information contained herein is suitable for user's intended purpose.

mdbs reserves the right to incorporate design improvements and new functions in its software products and software systems. Recent improvements may not always be reflected in documentation.

© Copyright 1987 Micro Data Base Systems, Inc.

TABLE OF CONTENTS

Page

CHAPTER I. OVERVIEW I-1
Additional DML Commands I-2

CHAPTER II. PAGE-IMAGE POSTING II-1
Command Details II-2
Example II-7

CHAPTER III. TRANSACTION LOGGING III-1
Command Details III-2
Recovery Utility - RCV III-9

CHAPTER IV. TRANSACTION LOG FILE IV-1
Transaction Log Functions IV-9

CHAPTER V. RCV MESSAGES V-1

CHAPTER VI. DMS COMMAND STATUS MESSAGES VI-1

© COPYRIGHT 1987 Micro Data Base Systems, Inc.

I. OVERVIEW

A vital component of a data base management system is a facility for recovering from various catastrophic happenings such as media failure, system failure, or human error. The RTL form of MDBS has been designed to provide special recovery capabilities through its transaction logging and page image posting features. The RTL form is a modified data base control system that includes a special group of DML commands for controlling the nature of transaction logging and page image posting that is desired during the execution of an application program. In some environments, the RTL form of MDBS III may require a larger page buffer than the standard form. Also supplied with the RTL form of MDBS is an interactive recovery utility called RCV.

A transaction has occurred whenever the data base contents are modified (creating, changing, or deleting data). RTL automatically logs all transactions made on a data base since the last data base back-up. When a media failure occurs, data base information on disk is destroyed. When this happens, RCV can be executed to automatically apply all logged transactions to the back-up copy of the data base. This restores the data base to a point close to the last completed transaction before the media failure occurred.

When a system failure (e.g., power interruption) occurs, information in the main memory buffers may be destroyed before the data base is properly closed. This means that the data base may be inconsistent, since some revised data base pages involved in a complex sequence of transactions may not yet have been flushed out to disk at the point of failure, while other revised pages involved in the complex transaction sequence have already been written to disk. RCV can be used to restore the data base to a consistent state. However, if a transaction was in the log file buffer at failure time, then that transaction is lost. Prior to invoking RCV it is prudent to make a back-up copy of the back-up copy of the data base. This can be time consuming for large data bases. Thus RTL provides an alternative defense against many system failures by allowing page image posting. This means that all pages revised during a complex transaction sequence are committed to disk only when that transaction is complete. System failure in the midst of a complex transaction will not result in data base inconsistency. Only the interrupted transaction has been lost when the data base is re-opened. Page image posting is not meaningful in multiuser processing situations, since all users share the same page buffers.

When human error occurs, the data base has been changed in an authorized (from the standpoints of security and feasibility restrictions) but erroneous manner. For instance, a data entry person may have entered transactions that were later discovered to be invalid. In such a situation it is important to be able to roll back the data base, effectively deleting the erroneous transactions. This selective restoration of a data base is accomplished with RCV, which allows the transaction log to be scanned to identify the erroneous transactions. RCV is then used to selectively apply only the valid transactions on the log file to the data base back-up, with the result that the effects of invalid transactions have been eliminated from the data base.

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS RTL MANUAL
Overview

ADDITIONAL DML COMMANDS

The RTL form of MDBS III supports all of the DML commands that are normally provided with the standard form. In addition, it offers seven other DML commands that are specifically related to recovery. Invoking a recovery command follows the same conventions as invoking any other DML command. The DML recovery commands fall into two groups:

- those that deal with page image posting of complex transactions

PIFD	page image file declaration
TRBGN	begin posting a complex transaction sequence
TRCOM	the posted transaction sequence is committed to the data base
TRABT	abort the complex transaction sequence

- those that deal with the logging of transactions

LGFILE	specify an alternative log file
TRBGN	start logging a complex transaction sequence
TRCOM	end the logging of a complex transaction sequence
TRSYNC	synchronizes the log file in multiuser environments
LGFLSH	flush the log buffer to the log file
LGMSG	write a message to the log file

When using RTL, all transactions are logged; thus a log file must be on-line, preferably on some device other than the one(s) containing the data base. The use of a page image file is optional. RTL commands other than LGFLSH and LGMSG are not available for IDML processing. However, when IDML is configured with RTL (rather than DMS), automatic transaction logging of all IDML transactions occurs. The IDML user can specify the log file name on the operating system command line, as described in the appropriate system specific manual.

Chapter II describes page image posting. Chapter III describes transaction logging and the recovery utility (RCV). The log file format is presented in Chapter IV. The final chapter, Chapter V, explains the messages that the recovery utility can issue while it is running.

© Copyright 1987 Micro Data Base Systems, Inc.

II. PAGE-IMAGE POSTING

Page-image posting may be considered to form a first line of defense against data base inconsistency due to system failures such as power interruption. The transaction logging commands provide the ultimate defense against failures of many kinds, including power failures, hardware malfunctions, and erroneous (although authorized) data modification. Page-image posting can **optionally** be used by an application program.

Page-image posting allows the application developer to specify the beginning of a complex transaction sequence (with the TRBGN command). The data base changes caused by the transaction sequence are not incorporated into the data base until the commit command (TRCOM) is invoked. Note that DBCLS performs an automatic TRCOM. A transaction sequence can be aborted by invoking the abort (TRABT) command. An abnormal termination of the program in the midst of a complex transaction sequence will not leave the data base inconsistent. When the data base is re-opened, it is consistent. It is current up to the point of the last invocation of TRBGN before the interruption.

A page image preservation file is used by MDBS.DMS to provide this automatic recovery capability. The page image file must be declared with the PIFD command prior to opening the data base. If a page image file of insufficient size is used, then the data base could become inconsistent if a transaction sequence is abnormally interrupted. In this event, a nonzero command status to that effect is returned when an attempt is made to re-open the data base. The log of transactions can be used to recover from such a situation (see Chapter III).

The four commands used to control page image posting within an application program are described in the next section. This chapter ends with a general example of application program flow when using page image posting.

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS RTL MANUAL
Page–image Posting

COMMAND DETAILS

PIFD — Page-Image File Declaration

PIFD

Command and Arguments PIFD,iblk

Currency Indicators

Used	Changed	Change Description
none	none	

Description

The fully qualified file name, indicated by **iblk**'s host language variable, is declared to be a page image preservation file for use by MDBS.DMS in page image posting. If posting is desired, this command must be invoked prior to use of the TRBGN, TRCOM, and TRABT commands. The host language variable must be consistent with a string data item. The size (in bytes) is operating system dependent and is documented in the corresponding system specific manual.

If PIFD is invoked, it must appear before the data base is opened.

Examples of Command Usage

```
block/direct     .. E0 = PIFD ("iblk")
block/indirect .. E0 = DMS ("PIFD,iblk")
record/direct    .. E0 = PIFD (iblk)
record/indirect .. E0 = DMSD ("PIFD",iblk)
```


© Copyright 1987 Micro Data Base Systems, Inc.

MDBS RTL MANUAL
Page–image Posting

TRABT — **TRansaction ABorT** — **TRABT**

Command and Arguments TRABT (no arguments)

Currency Indicators

Used	Changed	Change Description
none	most currency indicators are set to null	

Description

Posting Effects (occur only if PIFD has been invoked):
This command aborts a complex transaction sequence that was initiated by the TRBGN command. A transaction cannot be aborted after it has already been committed (with the TRCOM command). This command invokes NCI internally to ensure that no inconsistent currency indicators will be present.

Logging Effects:
The transaction sequence that is aborted will be ignored by the RCV recovery processor, but TRABT cannot be invoked if the application program does not use the page-image file.

Examples of Command Usage

```
block/direct   .. E0 = TRABT (    )
block/indirect .. E0 = DMS ("TRABT")
record/direct  .. E0 = TRABT (    )
record/indirect.. E0 = DMS ("TRABT")
```

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS RTL MANUAL
Page–image Posting

TRBGN

TRansaction BeGiN

TRBGN

Command and Arguments TRBGN (no arguments)

Currency Indicators

Used	Changed	Change Description
none	none	

Description

Posting Effects (occur only if PIFD has been invoked):
This command denotes the beginning of a complex transaction , a sequence of several DML commands that logically comprise a transaction to an end user.

Corresponding to the complex transaction beginning are the complex transaction abort and the complex transaction commit. The complex transaction abort (TRABT) signals the end of the transaction and that the DML commands within the complex transaction are not to become part the data base. The complex transaction commit (TRCOM) singals the end of the complex transaction and permanently applies the data base changes of the complex transaction to the data base.

After TRCOM or TRABT has been invoked, TRBGN must be re-invoked to initiate another complex transaction. If TRBGN is re-invoked without having issued a TRABT or TRCOM command, the data base control system returns a nonzero command status and ignores the re-invocation of TRBGN.

Logging Effects:
This command denotes the beginning of a complex transaction , a sequence of several DML commands that logically comprise a transaction to an end user.

The data base control system records all individual DML commands that change the data base and all complex transaction commands (begin, abort, and commit). During the data base recovery, however, RCV ignores any transactions that follow TRBGN unless a subsequent TRCOM command exists in the log file.

The TRBGN and TRCOM commands are used in tandem to ensure the data base recovery either performs the entire complex transaction (with all of the DML commands comprising the complex transaction) or ignores the complex transaction (if the TRCOM is not present).

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS RTL MANUAL
Page–image Posting

Examples of Command Usage

```
block/direct   .. E0 = TRBGN (     )
block/indirect .. E0 = DMS ("TRBGN")
record/direct  .. E0 = TRBGN (     )
record/indirect.. E0 = DMS ("TRBGN")
```

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS RTL MANUAL
Page–image Posting

TRCOM — TRansaction COMmit — TRCOM

Command and Arguments TRCOM (no arguments)

Currency Indicators

Used	Changed	Change Description
none	none	

Description

Posting Effects (occur only if PIFD has been invoked):
The complex transaction commit (TRCOM) signals the end of the complex transaction (a sequence of DML commands that modify the data base) and permanently applies the data base changes of the complex transaction to the data base. It has the added effect of clearing the contents of the page image file declared with the PIFD command.

Logging Effects:
This command indicates the end of a complex transaction. Any transactions logged since the last TRBGN command was invoked can now be processed by the recovery utility (RCV) while restoring the data base.

Examples of Command Usage

```
block/direct   .. E0 = TRCOM (     )
block/indirect .. E0 = DMS ("TRCOM")
record/direct  .. E0 = TRCOM (     )
record/indirect.. E0 = DMS ("TRCOM")
```


© Copyright 1987 Micro Data Base Systems, Inc.

MDBS RTL MANUAL
Page–image Posting

EXAMPLE

The following example illustrates a typical program flow when using page image posting:

```
pifd
.
.
.
dbopn
.
.
.
ov (obtain and validate new data from end user)
.
.
.
trbgn
.
.
.
(incorporate new data into data base with appropriate DML commands)
.
.
.
trcom
.
.
.
(iterate to ov, as desired)
dbcls
```

Notice that the page image file is declared before opening the data base. The end user is prompted for new data. A complex transaction sequence is initiated by TRBGN. A sequence of transactions is generally considered to be complex if it involves related changes to several records. When TRCOM is executed, all changes that make up the complex transaction sequence are committed to the data base. If the program is interrupted after TRBGN and before TRCOM is completed, the changes are not committed to the data base. Even though the data base is not closed in this situation, it can be reopened and its contents are consistent.

© Copyright 1987 Micro Data Base Systems, Inc.

III. TRANSACTION LOGGING

Transaction logging allows an application program to make use of the log file defined in the DDL specification. The name of the file used for logging a program's transactions can be changed with the LGFILE command. **All** transactions since the last data base back-up are automatically logged onto the log file. The log file is used by the RCV utility (provided with MDBS-RTL). RCV can be used to automatically re-apply transactions to an old back-up copy of the data base, thereby recreating an up-to-date data base. The user has extensive control over which transactions are re-applied.

The TRBGN command is used to indicate the start of a complex sequence of transactions, for the purpose of transaction logging. The TRCOM command terminates the complex transaction sequence started by TRBGN. RCV will ignore any logged transactions that occur after an TRBGN if a subsequent TRCOM was not reached during a run unit. Other commands that can be used in conjunction with transaction logging are LGFLSH and LGMSG which flush the log file buffer and insert messages into the log file, respectively.

MDBS.RTL can use any initialized MDBS III data base if the data base has an associated log file. To use an existing data base with RTL, follow these guidelines:

- Create a log file for the data base - Run RCV for the data base and specify the name of the new log file. Within the Recovery Utility, reset the log file. This creates a new log file with the specified name

- Revise application programs to use the log file - Within each application program, specify the log file before opening the data base

This chapter next describes the five recovery commands that can be used in conjunction with transaction logging and concludes by describing the data base recovery utility RCV, its command line arguments, and its functions.

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS RTL MANUAL
Transaction Logging

COMMAND DETAILS

TRBGN — TRansaction BeGiN

TRBGN

Command and Arguments TRBGN (no arguments)

Currency Indicators

Used	Changed	Change Description
none	none	

Description

Logging Effects

When this command is invoked, it indicates the beginning of a complex sequence of transactions. Although all transactions are automatically logged, RCV will ignore any transactions that follow TRBGN unless a subsequent TRCOM command is encountered. The TRBGN and TRCOM commands are used in tandem to ensure that either all or none (if a complex sequence is not completed) of the transactions, which form part of a complex sequence, are used by RCV during recovery.

Posting Effects (occur only if PIFD has been invoked):

This command denotes the beginning of a complex transaction sequence, involving the use of several DML commands. The changes that these DML commands cause in the data base do not become permanent until the TRCOM command is invoked. TRCOM commits these changes to the data base. The TRABT command aborts a complex transaction. After TRCOM or TRABT has been invoked, TRBGN must be re-invoked to initiate another complex transaction. If TRBGN is re-invoked without having issued a TRABT or TRCOM command, a nonzero command status is issued and the re-invocation of TRBGN is ignored by MDBS.DMS.

Examples of Command Usage

```
block/direct    .. E0 = TRBGN (    )
block/indirect  .. E0 = DMS ("TRBGN")
record/direct   .. E0 = TRBGN (    )
record/indirect .. E0 = DMS ("TRBGN")
```


© Copyright 1987 Micro Data Base Systems, Inc.

MDBS RTL MANUAL
Transaction Logging

TRCOM — TRansaction COMmit — TRCOM

Command and Arguments TRCOM (no arguments)

Currency Indicators

Used	Changed	Change Description
none	none	

Description

Logging Effects:

This command indicates the end of a complex sequence of transactions. Any transactions logged since the last TRBGN command was invoked can now be processed by the RCV utility for data base restoration.

Posting Effects (occur only if PIFD has been invoked):

This command commits a complex transaction, that was initiated by the TRBGN command, to the data base. It has the added effect of clearing the contents of the page-image file declared with the PIFD command.

Examples of Command Usage

```
block/direct     ... E0 = TRCOM (     )
block/indirect   ... E0 = DMS ("TRCOM")
record/direct    ... E0 = TRCOM (     )
record/indirect  ... E0 = DMS ("TRCOM")
```

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS RTL MANUAL
Transaction Logging

TRSYNC — TRansaction SYNChronization — TRSYNC

Command and Arguments TRSYNC,iblk

Currency Indicators

Used	Changed	Change Description
none	none	

Description

Logging Effects:

TRSYNC synchronizes the transaction log file in multiuser environments. The synchronization forces all complex transactions from all run units to complete before any new complex transactions can begin. While the synchronization takes place, all TRBGN commands receive command status 64, operation rejected due to log file synchronization, until all outstanding complex transactions have been committed.

iblk is consistent with a two byte integer variable that specifies when the multiuser data base control system synchronizes the log file. **iblk** data values have these effects:

- **iblk is positive** - MDBS.RTL automatically synchronizes the log file on a periodic basis. The time period between successive synchronizations is the value of **iblk**, specified in minutes. For example, an iblk value of 30 causes MDBS.RTL to synchronize the log file every half hour (approximately).

- **iblk is zero** - MDBS.RTL performs one synchronization immediately and turns off automatic synchronization.

- **iblk is negative** - MDBS.RTL turns off all synchronization.

TRSYNC can be invoked in several ways:

- **from an application program** - use the TRSYNC command

- **from DBMTR** - use the DBMTR menu; TRSYNC runs with its parameter set to zero

- **from the multiuser data base control system** - use the command argument **s** followed immediately by the time interval

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS RTL MANUAL
Transaction Logging

Posting Effects (occur only if PIFD has been invoked):

TRSYNC has no effects on posting.

Examples of Command Usage

```
block/direct    .. E0 = TRSYNC (     )
block/indirect  .. E0 = DMS ("TRSYNC")
record/direct   .. E0 = TRSYNC (     )
record/indirect .. E0 = DMS ("TRSYNC")
```

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS RTL MANUAL
Transaction Logging

LGFILE — LoG FILE specification — LGFILE

Command and Arguments LGFILE,iblk

Currency Indicators

Used	Changed	Change Description
none	none	

Description

The fully qualified name of a file to which transactions can be logged is specified by iblk's host language variable. This variable must be consistent with a string data item whose maximum length is operating system dependent (see the appropriate system specific manual for size information). If the file to which transactions are to be logged is the same as the log file specified in the DDL specification, this command need not be invoked. If it is invoked, this command must appear before the DBOPN command.

Examples of Command Usage

```
block/direct    .. E0 = LGFILE ("iblk")
block/indirect  .. E0 = DMS ("LGFILE,iblk")
record/direct   .. E0 = LGFILE (iblk)
record/indirect .. E0 = DMSD ("LGFILE",iblk)
```


© Copyright 1987 Micro Data Base Systems, Inc.

MDBS RTL MANUAL
Transaction Logging

LGFLSH

LoG FLuSH

LGFLSH

Command and Arguments **LGFLSH (no arguments)**

Currency Indicators

Used	Changed	Change Description
none	none	

Description

The buffers holding transactions to be logged are flushed to the log file when this command is invoked. LGFLSH is automatically invoked by DBSAVE. This is especially useful when an important transaction has been processed by the system. It can be written to the log file before the log file buffer is full.

Examples of Command Usage

```
block/direct   .. E0 = LGFLSH (     )
block/indirect .. E0 = DMS ("LGFLSH")
record/direct  .. E0 = LGFLSH (     )
record/indirect.. E0 = DMS ("LGFLSH")
```

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS RTL MANUAL
Transaction Logging

LGMSG — LoG file MeSsaGe — LGMSG

Command and Arguments **LGMSG,iblk**

Currency Indicators

Used	Changed	Change Description
none	none	

Description

The message specified in iblk's host language variable is written to the log file. This variable must be consistent with a string data item not exceeding 90 bytes. Messages written to the log file can be listed using the RCV utility program. The use of the LGMSG command is especially useful for surveillance.

Examples of Command Usage

```
block/direct    .. E0 = LGMSG ("iblk")
block/indirect  .. E0 = DMS ("LGMSG,iblk")
record/direct   .. E0 = LGMSG (iblk)
record/indirect .. E0 = DMSD ("LGMSG", iblk)
```


© Copyright 1987 Micro Data Base Systems, Inc.

MDBS RTL MANUAL
Transaction Logging

RECOVERY UTILITY - RCV

The RCV processor enables you to display logged transactions and to apply these transactions to a back-up copy of the data base. RCV enforces all security conditions specified in the DDL. You can display transactions which involve a data item, record type, or set for which you have read access; you can only apply transactions (to the data base) for which you have write access to all data items, record types, and sets involved in the transactions.

Before using RCV, take these precautions:

- *MAKE A COPY OF YOUR BACK-UP DATA BASE*

  If you are using RCV, your master data base has been lost or damaged in some way. This increases the importance of the information in your back-up data base. Do not risk losing it during the recovery process.

- *MAKE A COPY OF YOUR LOG FILE*

  The RCV processor does not write to the transaction log file, but a back-up is cheap insurance.

MDBS.RCV is packaged as an executable file. For most operating systems, you invoke RCV by entering the name of the executable file and any appropriate command line arguments. Also, make sure the main area of your data base is on-line. The precise syntax to invoke RCV is in your MDBS III System Specific Manual, chapter IX.

Command Line Arguments

MDBS.RCV has several optional command line arguments that control things such as the amount of workspace for data base page buffers and the operating mode (interactive or batch) of RCV. The following command line arguments are available for RCV:

–b*nnnnn*

This argument explicitly allocates the page buffer region. ***nnnnn*** is the (decimal) number of bytes to be allocated for the page buffer region. This number should be at least as large as the "minimum DMS buffer region size" reported by the DDL Analyzer; otherwise DMS command status 31 results.

If this argument is not specified, RCV automatically allocates approximately half of the available memory. RCV reserves the remainder of its workspace for items such as file control blocks, a stack, and sort work areas.

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS RTL MANUAL
Transaction Logging

If too large a page buffer region is requested, a message indicating "excessive memory request" is displayed. The remaining memory reserved for non-data base working space is normally sufficient. However, in rare cases this working space may be insufficient, resulting in various kinds of situations where the data base may be left open. In such a case, you can specify a smaller page buffer region with the **–b** argument to allow a larger non-data base working space.

–d*dbname*

The **–d** argument specifies the data base to use with RCV. ***dbname*** is the fully qualified, operating system file name of the main data base area. If this argument is omitted from the command line, RCV prompts the user for this information.

–f This argument specifies a fast update. RCV updates all transactions in the log file and uses some efficiencies while it updates the data base.

–i*filename*

The **–i** argument specifies that RCV operates in batch mode and is to perform the commands within the file ***filename***. ***filename*** is an ASCII text file that consists of the log file name, menu selections, and other responses to RCV prompts. It is a keystroke file.

–m The **–m** option suppresses (mutes) the display of the initial banner message for RCV. This option is intended primarily for original equipment manufacturers that call RCV from another program.

–o*outfile*

The **–o** argument causes RCV to perform the outfile option. ***outfile*** is the name of the new log file. See the next section for a description of the outfile option.

–p*password*

The **–p** argument specifies the data base password for the user. As with the arguments for the data base name and user, if this argument is omitted from the command line, RCV prompts you for a valid password before providing access to the data base.

–u*username*

This argument specifies the name of the person to access the data base. ***username*** is any valid user defined for the data base. (The user name is defined in the user section of the DDL source text or is defined in a session with DMU.) If the **–u** argument is omitted from the command line, RCV prompts you for a valid user name.

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS RTL MANUAL
Transaction Logging

-v This argument designates RCV to be verbose, which means that RCV provides a running commentary of its actions as it processes a log file.

When you invoke the RCV module, RCV displays its banner message on the console screen (Figure III-1), unless the -m argument has been specified on the command line.

```
MDBS RCV Ver 3.09a
(C) Copyright 1981, 1986, Micro Data Base Systems, Inc.
Lafayette, IN  47902
Serial Number:  xxxxxxxx
        Data base:  JOBS.DB
        User name:  D LEHR
        password:
```

Figure III-1. RCV Banner Message and Initial Prompts

Before permitting access to the data base, the system prompts for the file name of the main data base area, the user name, and password. The password is not echoed to the console when it is entered. Command line arguments d, u, and p provide an alternative means of specifying this information and suppress these prompts when RCV is invoked.

If you provide valid user and password information for the data base, RCV performs the following tasks:

- prompts you for the log file name. To use the default log file (specified in the DDL source text) for the data base, press the return key
- displays the main RCV menu (Figure III-2)
- prompts you for a menu selection

```
Option Summary:
        List:   List transactions from the log file
        Quit:   Quit and return to operating system
        Reset:  Reset the log file
        Scan:   Scan the log file
        Update: Update the data base
        Combine:Combine log files
        Outfile:Update data base and create new log file

OPTIONS: (List, Quit, Reset, Scan, Update, Combine, Outfile)?
```

Figure III-2. RCV Main Menu and Option Prompt

After performing a selected option, RCV returns to the main menu and prompts you for the next option. These options are described below.

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS RTL MANUAL
Transaction Logging

List of the log file (L)

The list option displays all transactions in the log file within the indicated range of transaction numbers. The data values involved in transactions are not displayed by this command. This command is useful for determining the starting and ending transactions to be used in recovery processing.

When this option is selected, you are prompted for the range of transactions to be listed:

First transaction # [1]?
Last transaction # [highest]?

Transaction numbers are entered as decimal numbers. Pressing a carriage return in response to the first transaction number has the same result as entering 1. Pressing a carriage return in response to the last transaction number has the same effect as entering the number of last transaction on the log file.

When the transaction range has been specified, the operator is presented with:

User name [all]?

This allows the operator to list only that information which pertains to the specified user. The user name which is entered must be a name which appears in "User Login" transactions. If the default is entered (i.e., the RETURN or ENTER key is pressed), information pertaining to all users within the range is listed.

The format of output from the list option is as follows:

run unit id transaction number .. transaction description

The transaction number is a unique number that identifies the transaction. The run unit id is a number assigned by RTL to a run unit; the meaning of this number is system specific and is generally of little interest. A detailed description of each transaction type along with its data format appear in chapter IV, Transaction Log File.

Quit (Q)

When this option is selected, the data base and log file are closed. Control is returned to the operating system.

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS RTL MANUAL
Transaction Logging

Reset the log file (R)

Exercise caution when using this command; it destroys the contents of the log file.

When the reset option is selected, RCV prompts you to assure that the reset is actually desired. The reset option has the effects of deleting all information from the indicated log file and re-initializing that file. You may wish to make an archival copy of the log file before the reset option is used.

Scan the log file (S)

In scanning the log file, RCV displays all messages, data base opens, data base closes, user logins, and user logouts appearing on the log file within the indicated range of transaction numbers. The user name associated with each login is also displayed.

When this option is selected, you are prompted for the range of transactions to be scanned:

```
First transaction # [1]?
Last transaction # [highest]?
```

Transaction numbers are entered as decimal numbers. Pressing a carriage return in response to the first transaction number has the same result as entering 1. Pressing a carriage return in response to the last transaction number has the same effect as entering the number of last transaction on the log file.

When the transaction range has been specified, the operator is presented with:

```
User name [all]?
```

This allows the operator to scan only that information which pertains to the specified user. The user name which is entered must be a name which appears in "User Login" transactions. If the default is entered (i.e., the RETURN or ENTER key is pressed), information pertaining to all users within the range is scanned.

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS RTL MANUAL
Transaction Logging

Update the data base (U)

The update option automatically applies all log file transactions within the indicated range for the indicated user(s) to the copy of the back-up data base.

When this option is selected, RCV prompts you for the names of two temporary work files that are needed by RCV to update the data base.

```
Scratch File #1 (SCRATCH1.TMP)?
Scratch File #2 (SCRATCH2.TMP)?
```

Next, RCV prompts you for the range of transactions that are to be used for updating the copy of the back-up data base:

```
First transaction # [1]?
Last transaction # [highest]?
```

Transaction numbers are entered as decimal numbers. Pressing a carriage return in response to the first transaction number has the same result as entering 1. Pressing a carriage return in response to the last transaction number has the same effect as entering the number of last transaction on the log file.

When the transaction range has been specified, the operator is presented with:

```
User name [all]?
```

This allows the operator to update using only those transactions in the range that pertain to the specified user. The user name which is entered must be a name which appears in "User Login" transactions. If the default is entered, transactions pertaining to all users within the range are used for updating.

Combine log files (C)

The combine option permits you to concatenate successive log files for a data base. You can save off a log file after completing a certain amount of work (such as at the end of the day) and use a new log file to record subsequent data base changes. If the data base becomes damaged before the next back-up occurs, concatenate the log files with this option and then update the data base.

To concatenate log files, invoke RCV and specify the first log file as the log file to use with RCV.

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS RTL MANUAL
Transaction Logging

When this option is selected, RCV repeatedly uses the following message to prompt you for the names of successive transaction files. Enter the next transaction file name and press return; when you have no more transaction file names to enter, press the return key as your response. This completes the combine option.

Next Transaction File?

After combining the transaction log files, you can update the data base by performing the update option with the composite log file.

Outfile: Update data base and create new log file

The outfile option performs two functions:

- it updates the data base from the current log file
- it updates the log file contents

The data base update is identical to the update option mentioned earlier.

Discrepancies between the data base key values in the log file and the actual location of the record occurrences result from updating the data base. As RCV updates a data base, it uses the data base key values from the log file to locate record occurrences. As a new record is created or an existing record is modified, RCV may change the location of that record, rendering the log file entry inconsistent for any additional transactions for that record.

The outfile option eliminates the discrepancies in the log file by systematically editing the data base keys (the location of the record occurrences within the data base) to be consistent with the location of the records in the data base. By updating the data base keys in the log file, you can use the updated log file with the updated data base to continue processing data base transactions; you do not have to make an additional back-up of the data base.

When this option is selected, RCV prompts you for several pieces of information. RCV first prompts for the name of the updated log file with the following message:

Enter output file name:

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS RTL MANUAL
Transaction Logging

The next two prompts specify the names of two temporary work files that are needed by RCV to update the data base.

Scratch File #1 (SCRATCH1.TMP)?
Scratch File #2 (SCRATCH2.TMP)?

RCV needs to know the specific transactions to apply to the data base. The next two prompts specify the range of transactions that are to be used for updating the copy of the back-up data base:

First transaction # [1]?
Last transaction # [highest]?

Transaction numbers are entered as decimal numbers. Pressing a carriage return in response to the first transaction number has the same result as entering 1. Pressing a carriage return in response to the last transaction number has the same effect as entering the number of last transaction on the log file.

When the transaction range has been specified, you are is presented with:

User name [all]?

This allows you to update the data base and log file using only those transactions in the range that pertain to the specified user. The user name which is entered must be a name which appears in "User Login" transactions. If the default is entered, transactions pertaining to all users within the range are used for updating.

When this option completes, RCV returns to its main menu and prompts you for your next selection.

© Copyright 1987 Micro Data Base Systems, Inc.

IV. TRANSACTION LOG FILE

MDBS.RTL has twenty-three types of data base transactions that it can record in a log file. This chapter lists and describes each of these transaction types. The listing for a single transcation includes the following information:

- RCV Message - This is the transaction message RCV displays when it encounters the transaction
- Description - transaction synopsis
- DMS Commands - DMS Commands that can generate this transaction type
- Function Number - the function number stored in the log file
- Record Length - the number of bytes this transaction uses in the log file
- Record Layout - shows how the record parameters are arranged in the record occurrence

The transaction file consists of a byte stream in which transactions immediately follow one another; there are no separator characters. The first byte of each logical disk record contains a hex A7, except that the last block contains a hex A8 value, indicating an end of file. The second byte of each record is a checksum such that the sum of all bytes in the record is zero. There is an 'xx' byte header at the beginning of the file.

Abort Transaction Sequence

Appearance of this message indicates a successful invocation of TRABT in a source program. Any transactions between the preceding "Begin Transaction Sequence" and the abort will not be recovered.

DMS Commands	Function Number	Record Length	---Record Layout--- bytes	description
TRABT	20	2	1	Run Unit ID

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS RTL MANUAL
Transaction Log File

Begin Transaction Sequence

Appearance of this message indicates that the user has called the TRBGN command in a source program. Any further transactions for this user will **not** be processed by RCV unless an TRCOM transaction or a user/system logoff transaction is encountered.

DMS Commands	Function Number	Record Length	---Record Layout--- bytes	description
TRBGN	16	2	1	Run Unit ID

Boolean Operation

This message indicates that the application performed a Boolean command. The log file entry records the contents of user indicator one, user indicator two, and the set names used in the Boolean command.

DMS Commands	Function Number	Record Length	---Record Layout--- bytes	description
AMM	23	13+	1	Run Unit ID
AMO			2 - 3	Data Length
AOM			4 - 7	User Indicator 1
AOO			8 - 11	User Indicator 2
XMM			12	Boolean Command Type
XMO			13+	Set Names
XOM				
XOO				

Commit Transaction Sequence

This transaction indicates the end of a transaction sequence.

DMS Commands	Function Number	Record Length	---Record Layout--- bytes	description
TRCOM	15	2	1	Run Unit ID

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS RTL MANUAL
Transaction Log File

Create Clustered Record

This function creates a new record occurrence that is close to another record in the data base; generally, the data base control system tries to create the new record on the same page as the clustering record.

The type of the record is listed, followed by the identifier of the newly created record. The clustering record identifier specifies the data base key of the existing record occurrence (that the new record is clustered around). Note that if the record is the auto member or owner of any sets, several "Insert Record in Set" transactions may follow.

DMS Commands	Function Number	Record Length	---Record Layout--- bytes	description
CRS	22	13+	1	Run Unit ID
			2 - 3	Data Length
			4	Record Type
			5 - 8	Record Identifier
			9 - 12	Clustering Record Identifier
			13+	Data Values

Create Record

The record type of the record is listed, followed by the identifier of the newly created record. Note that if the record is the auto member or owner of any sets, several "Insert Record in Set" transactions may follow.

DMS Commands	Function Number	Record Length	---Record Layout--- bytes	description
CRS	1	9+	1	Run Unit ID
CRA			2 - 3	Data Length
			4	Record Type
			5 - 8	Record Identifier
			9+	Data Values

Data Base Closed

No additional information is listed. Note that for a multi-user system, this indicates the close of the data base. A run unit close is indicated by a **User Logoff**.

DMS Commands	Function Number	Record Length	---Record Layout--- bytes	description
DBCLS	18	1		None

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS RTL MANUAL
Transaction Log File

Data Base Opened

No additional information is listed.

DMS Commands	Function Number	Record Length	---Record Layout--- bytes	description
DBOPN	19	1		None

Delete Record

The record identifier of the deleted record is listed.

DMS Commands	Function Number	Record Length	---Record Layout--- bytes	description
DRC DRM DRO	12	6	1 2 - 5	Run Unit ID Record Identifier

Insert Record in Set

The name of the set is listed, followed by the identifier of the record which is being added to the set. For certain classes of sets, the identifier of the current owner (member) of the set is displayed, followed by the identifier of the current member (owner).

Transaction function 2 inserts a record as a set member into a system owned set whose members are ordered IMMATERIAL, FIFO, LIFO, or SORTED.

DMS Commands	Function Number	Record Length	---Record Layout--- bytes	description
IMS	2	8	1 2 - 3 4 - 7	Run Unit ID Set Type Record Identifier

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS RTL MANUAL
Transaction Log File

Insert Record in Set (continued)

Transaction function 3 inserts a record as a set member into a system owned set whose members are ordered NEXT or PRIOR.

DMS Commands	Function Number	Record Length	---Record Layout--- bytes	description
IMS	3	12	1	Run Unit ID
			2 - 3	Set Type
			4 - 7	Record Identifier
			8 - 11	Adjacent Record Identifier

Transaction function 4 inserts a record as a set member (or set owner) into a non-system owned set whose members (or owners) are ordered IMMATERIAL, FIFO, LIFO, or SORTED.

DMS Commands	Function Number	Record Length	---Record Layout--- bytes	description
IMS	4	12	1	Run Unit ID
IOS			2 - 3	Set Type
			4 - 7	Record Identifier
			8 - 11	Owner/Member Identifier

Transaction function 5 inserts a record as a set member into a non-system owned set whose members are ordered NEXT or PRIOR.

DMS Commands	Function Number	Record Length	---Record Layout--- bytes	description
IMS	5	16	1	Run Unit ID
			2 - 3	Set Type
			4 - 7	Record Identifier
			8 - 11	Member Identifier
			12 - 15	Adjacent Record Identifier

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS RTL MANUAL
Transaction Log File

Insert Record in Set (continued)

Transaction function 6 inserts a record as a set owner into a non-system owned set whose owners are ordered NEXT or PRIOR.

DMS Commands	Function Number	Record Length	---Record Layout--- bytes	description
IOS	6	16	1	Run Unit ID
			2 - 3	Set Type
			4 - 7	Record Identifier
			8 - 11	Owner Identifier
			12 - 15	Adjacent Record Identifier

Message

A message entered in the transaction file by the LGMSG command is displayed.

DMS Commands	Function Number	Record Length	---Record Layout--- bytes	description
LGMSG	17	4+	1	Run Unit ID
			2 - 3	Message Length
			4+	Message

Put Data in Record

The record type is displayed, followed by the record identifier of the record.

DMS Commands	Function Number	Record Length	---Record Layout--- bytes	description
PUTC	8	9+	1	Run Unit ID
PUTM			2 - 3	Data Length
PUTO			4	Record Type
			5 - 8	Record Identifier
			9+	Data

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS RTL MANUAL
Transaction Log File

Put Field in Record

The record type is displayed, followed by the item type. The record identifier of the record to be stored will also be displayed.

DMS Commands	Function Number	Record Length	---Record Layout--- bytes	description
PFC	7	11	1	Run Unit ID
PFM			2 - 3	Data Length
PFO			4 - 5	Item Type
			6	Record Type
			7 - 10	record Identifier
			11+	Data

Remove Record From Set

The name of the set, the current owner record's identifier and the current member's identifier are displayed.

DMS Commands	Function Number	Record Length	---Record Layout--- bytes	description
RMS	9	12	1	Run Unit ID
ROS			2 - 3	Set Type
			4 - 7	Owner Identifier
			8 - 11	Member Identifier

Remove Set Members

The set name and its current owner identifier are displayed.

DMS Commands	Function Number	Record Length	---Record Layout--- bytes	description
RSM	10	8	1	Run Unit ID
			2 - 3	Set Type
			4 - 7	Owner Identifier

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS RTL MANUAL
Transaction Log File

Remove Set Owners

The set name and its current owner identifier are displayed.

DMS Commands	Function Number	Record Length	---Record Layout--- bytes	description
RSO	11	8	1	Run Unit ID
			2 - 3	Set Type
			4 - 7	Member Identifier

User Login

This transaction indicates that a specific user has been using the data base.

DMS Commands	Function Number	Record Length	---Record Layout--- bytes	description
DBOPN	13	5+	1	Run Unit ID
			2 - 4	External Run Unit ID
			5+	User Name

User Logout

This transaction indicates that a specific user has ended use of the data base.

DMS Commands	Function Number	Record Length	---Record Layout--- bytes	description
DBCLS	14	2	1	Run Unit ID

No Operation

This is a null transaction; nothing happened.

DMS Commands	Function Number	Record Length	---Record Layout--- bytes	description
-	0	1		None

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS RTL MANUAL
Transaction Log File

TRANSACTION LOG FUNCTIONS

This section lists the functions of the transaction log by the function number. For more descriptive information about a function, look up the corresponding RCV message in the previous section.

Function Number	RCV Message	Record Length
0	**No Operation**	1
1	**Create Record**	9+
2	**Insert Record in Set**	8
3	**Insert Record in Set**	12
4	**Insert Record in Set**	12
5	**Insert Record in Set**	16
6	**Insert Record in Set**	16
7	**Put Field in Record**	11
8	**Put Data in Record**	9+
9	**Remove Record From Set**	12
10	**Remove Set Members**	8
11	**Remove Set Owners**	8
12	**Delete Record**	6
13	**User Login**	5+
14	**User Logout**	2
15	**Commit Transaction Sequence**	2
16	**Begin Transaction Sequence**	2
17	**Message**	4+
18	**Data Base Closed**	1
19	**Data Base Opened**	1
20	**Abort Transaction Sequence**	2
22	**Create Clustered Record**	13+
23	**Boolean Operation**	13+

© Copyright 1987 Micro Data Base Systems, Inc.

V. RCV MESSAGES

***** CAN'T CREATE FILE 'name'**

The specified file could not be created.

***** CAN'T OPEN FILE 'name'**

The specified file could not be opened.

***** DMS ERROR n DETECTED**

This message is usually caused by the same factors as the UPDATE ERROR FOUND message. Note that "n" corresponds to a data base error number.

***** IMPROPERLY EMBEDDED TRANSACTION SEQUENCE**

Two TRBGN transactions were encountered without any intervening TRABT or TRCOM transactions.

***** INSUFFICIENT MEMORY**

RCV does not have sufficient memory available to allocate needed buffers and tables. The pointers described in the installation information may be altered to increase the memory available.

***** INVALID TRANSACTION TYPE ENCOUNTERED**

An invalid transaction was encountered in the transaction file. This is usually due to a damaged transaction file. An end of file is assumed at the error point in the file, and processing may continue.

***** I/O ERROR ON FILE 'name'**

An Input/Output error was detected on the specified file.

***** RUN UNIT ID NOT FOUND**

For multi-user systems, a compact user id is maintained in the transaction file. If a transaction with an invalid or unknown user id is encountered, this message is displayed. Usually this will only happen if a User Login function has not been encountered for the user, which is often the result of an improper specification of the first transaction number.

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS RTL MANUAL
RCV Messages

*** SYSTEM CLOSE WITHOUT PREVIOUS OPEN

This message may come about due to a damaged transaction file or due to use of a first transaction specification of a transaction which is not a "system open" transaction.

*** SYSTEM OPENED BUT NOT PREVIOUSLY CLOSED

This rather unusual message indicates that two "system open" transactions without an intervening "system close" transaction were encountered. This may indicate a damaged transaction file.

*** UPDATE ERROR FOUND

This message may indicate several problems. Generally, it occurs if the data base and transaction file do not correspond (i.e., using a transaction file associated with data base "A" won't work if you are applying it to data base "B").

*** USE HEX NOTATION

This message is generated when the user fails to enter a valid hex number for a RCV prompt.

*** USER LOGIN NOT PREVIOUSLY LOGGED

In a multi-user system, two "user login" transactions have been encountered without an intervening "user logout" transaction.

© Copyright 1987 Micro Data Base Systems, Inc.

VI. DMS COMMAND STATUS MESSAGES

There are several command status messages that are exclusively related to the posting and logging as described in Chapters II and III:

****55**** **Log file I/O error** ****55****

Possible Causes:

1. An I/O error was encountered when trying to open the log file.
2. See also Command Status 51 (in the MDBS DMS Manual).

****56**** **Log file not current** ****56****

Possible Causes:

1. The log file is not up-to-date.

****57**** **Logging inhibited** ****57****

Possible Causes:

1. An I/O error was detected on the log file.
2. The log file may already be open.

****58**** **Page-image file I/O error** ****58****

Possible Causes:

1. An I/O error was encountered when an attempt was made to read from or write to the page-image preservation file.
2. See also Command Status 51 (in the MDBS DMS Manual).

****64**** **Operation Rejected due to Log File Synchronization** ****70****

Possible Causes:

1. A run unit attempted to start a new complex transaction (with TRBGN) while the data base control system was in the middle of synchronizing the transaction log file.

****70**** **No transaction in progress** ****70****

Possible Causes:

1. An attempt was made to abort or commit a transaction sequence without having previously invoked TRBGN.

© Copyright 1987 Micro Data Base Systems, Inc.

MDBS RTL MANUAL
DMS Command Status Messages

****71**** **Transaction already in progress** ****71****

Possible Causes:

1. TRBGN was invoked twice without having invoked TRABT or TRCOM between the two TRBGNs.

****72**** **Page-image posting not active** ****72****

Possible Causes:

1. TRABT was invoked without having previously declared a page- image preservation file with PIFD.

****73**** **Page-image file index overflow** ****73****

Possible Causes:

1. A transaction sequence was too long.
2. More than sixty pages have been altered in the course of a transaction sequence.

****74**** **Log file not present** ****74****

Possible Causes:

1. Using the RTL form of MDBS, the data base was opened without the log file being present. Processing continues.

© Copyright 1987 Micro Data Base Systems, Inc.

DOCUMENTATION COMMENT FORM

MDBS Document Title: _____ Version: _____

We welcome and appreciate all comments and suggestions that can help us to improve our manuals and products. Use this form to express your views concerning this manual.

Please do not use this form to report system problems or to request materials, etc. System problems should be reported to MDBS by phone or telex, or in a separate letter addressed to the attention of the technical support division. Requests for published materials should be addressed to the attention of the marketing division.

Sender:

_____ (name)	_____ (company)
_____ (position)	_____ (address)
_____ (telephone)	_____ (address)
	_____ (city, state, zip)

COMMENTS:

Areas of comment are general presentation, format, organization, completeness, clarity, accuracy, etc. If a comment applies to one or more specific pages, please cite the page numbers.

_____

_____

_____

_____

_____

_____

_____

Continue on additional pages, as needed. Thank you for your response.

© COPYRIGHT 1987 Micro Data Base Systems, Inc.